IN THE SPECIFICATION

On page 1, line 21 through page 2, line 17, replace the entire affected paragraphs with the paragraphs below, which show the changes made.

Current algorithms produce warnings or an indication of a conflict that interrupt the scheduling process. Present scheduling systems, known to one skilled in the art, use modal interfaces to detect and inform the user of errors. Errors that are detected in midstream of the user' work forcibly interrupt the user with a warning message. This warning message prevents the user from continuing work without at least acknowledging its presence. These warning mechanisms are rigid in their application and, if disabled, are completely ineffective at providing any type of failure notification. For example, this This warning process is accomplished through a series of programmed communications between the error detection program and the primary scheduling program that results in the scheduling operations ceasingto cease.

The prior art fails to provide an unobtrusive or concurrent means to notify a user of a resource scheduling conflict. By forcing the user to interact with a warning or notification of a pending error, the efficiency of the scheduling process suffers. If the user elects to dismiss the warning, no further indication is present to convey to the user that the conflict is still present. Furthermore, should a user elect to correct the conflict upon receiving notification, the user is required to navigate independently in the program to the position in the program that contains the cause of the conflict, which is both inefficient and costlyroot cause to the conflict, again decreasing efficiency and rising the cost of the scheduling method.

On page 5, lines 1-2, replace the entire paragraph with the paragraph below, which shows the changes made.

Figure 9 is a flow diagram illustrating a process of concurrent real-time error identification according to an embodiment.

On page 13, lines 1-13, replace the entire paragraph with the paragraph below, which shows the changes made.

Figure 5 is a Calendar page, 520 using one embodiment of a method for identification and notification of resource conflicts where conflicts are associated with specific resources. The depiction shows a schedule calendar for the date Monday, January 29, 2001. Under the date, 530, is a plurality of columns including one categorizing the resource being scheduled as "employee"~~employee~~. In the embodiment illustrated in figure 5 there are ten <u>resources 540, all of the category "employee"</u>~~resources all of the category employee, 510~~. To the right of the category listing is another column that individually identifies each resource. This column contains <u>employee identification numbers 550, which include</u> ~~an employee identification number illustrated in this embodiment by~~ empl#1 through empl#10, 550. In this embodiment, for the resources identified as empl#1, 560, empl#2, 570, and empl#4, 580, an exclamation <u>mark 590 is</u> ~~mark, 590,~~ located in the column immediately adjacent to the <u>employee</u> ~~employee's~~ identification number. This exclamation mark <u>590</u> indicates that there is a resource conflict associated with each of ~~these~~ employees <u>560, 570 and 580</u>.

On page 13, line 14, line 14 through page , replace the entire paragraph with the three paragraphs below, which shows the changes made.

An additional embodiment further allows the user to review all the conflicts associated with a particular resource and individually select which conflicts are to be resolved. **Figure 6** is a Conflicts page, 620, of an embodiment of the method for

identification and notification of <u>resource</u> ~~resources~~ conflicts <u>in which</u> ~~where~~ an

employee has been associated with a conflict and the user has <u>chosen</u> ~~selected~~ to review

the conflicts. After the user clicks on the associated symbol, a list of conflicts associated

with that resource is displayed, 630. In Figure 6 empl#1 has three rule based conflicts

630 ~~, associated with empl#1's involvement with the schedule~~.

Figure 7 presents another embodiment ~~of a Conflicts page, 620, showing~~

<u>including</u> a <u>Calendar page 720. Shown on the Calendar page 720 is a</u> calendar conflict,

730, associated with empl#2. As depicted in Figure 7, the calendar conflict is resolvable

by selecting the button on the bottom of the page, 740. By selecting the resolution

<u>button</u> ~~option~~, 740, one embodiment opens a new window on top of the conflict window

to offer possible solutions <u>and also to present</u> ~~as well as presenting~~ links to the area of the

program where alterations to the resource's utilization <u>can be made</u> ~~must be changed~~.

Figure 8 depicts a Resolve Calendar Conflict page, 820, <u>which gives</u> ~~giving~~ the

user an interactive means to correct the conflict. The conflict is repeated to the user,

830, and then, in this embodiment, two resolution choices are presented to the user, 840

and 850. In one embodiment, the user is presented with a solution that marks emlp#2 as

not attending the meeting thus removing the calendar conflict, 840. <u>The</u> ~~Secondly, the~~

user may <u>also</u> ignore the conflict until a new schedule is created, 850. Having made a

choice, the user than can apply the selection, cancel the resolution process, or seek

additional help by using the three interactive <u>"Apply, Cancel, and Help"</u> buttons located

at the bottom of the page, 860.

On page 14, line 15 through page 16, line 15, replace the entire affected

paragraphs with the paragraphs below, which show the changes made.

~~**Figure 9** is a high-level flow diagram of one embodiment of a method for~~

~~concurrent real-time identification of resource scheduling conflicts. The identification~~

~~process begins at 202 when a user activates a scheduling window or enters new data into~~

~~the resources scheduling window. In one embodiment, and error identification method~~

is located within the primary resource scheduling process and acts concurrently with the primary scheduling process to analyze the new data for any conflicts or problems. While error identification and notification of resource conflicts can operate independently of the primary resource scheduling process, it is shown here, 204, as an integral part of the overall scheduling system. If the analysis determines that the new data does not present any conflicts or problems a green light is displayed, 208.

If a conflict is recognized, one embodiment queries the analysis program to determine if the conflict is associated with a specific resource such as an employee, 209. If the conflict is resource specific the method further determines if the conflict is rule or calendar based, 210. If the conflict is rule based, the method directs the scheduling process to display a blue exclamation point next to the resource causing the conflict, 211. If the conflict is calendar based a red exclamation mark, in one embodiment, is displayed next to the resource causing the conflict, 213.

Should the examination identify a problem in the schedule, 206, and the problem is not associated with a specific resource or the resource that is identified is already marked, the method questions if the conflict has been suppressed, 214. A negative answer displays a red light, 214, confirming the existence of an unsuppressed scheduling conflict. Should the answer to the suppression question be affirmative, a query occurs as to the existence of any other unsuppressed conflicts, 215. If the answer is yes then, as indicated in 218, both a red light and a yellow light are illuminated indicating the presence of both an unsuppressed and a suppressed conflict. If only suppressed conflicts are present, then only a yellow light is illuminated, 216. At this point the user must elect to click on the displayed lights indicated in blocks 217, 216, or 218 for further information. If the user clicks on the red light, 220, a new window containing an HTML page is generated, 222, describing the conflict and suggesting potential solutions, 232. To do this, one embodiment utilizes a warning template files, 224, and the Windows Registry, 226.

Likewise if the user clicks on the yellow indication light, 228, an HTML page is generated, 230, describing the suppressed conflict or conflicts and potential solutions, 232. Contained within the new HTML page is a hyperlink that connects the user to the

~~specific window within the primary scheduling resource software where the conflicted resource must be altered. If the user clicks on the hyperlink, 234, one embodiment communicates with the primary resource scheduling software, 236, causing the display of the specific window, 240, necessary to modify the erroneous data. Additionally, the data that must be altered is highlighted to facilitate the user in resolving the conflict. If the user elects to suppress the conflict, 242, the conflict is marked appropriately, 244, and is stored in the Windows Registry, 248. If the user elects to un-suppress the conflict, 244, an HTML page is generated, 246 based on the new unsuppressed conflict.~~

Figure 9 is a flow diagram illustrating a process of concurrent real-time error identification according to an embodiment. The process begins at 902 when a user of a scheduling system enters new data. In one embodiment, an error identification method operates within and concurrent with the primary resource scheduling process and analyzes data for any conflicts. While error identification and notification of resource conflicts can operate independently of the primary resource scheduling process, in one embodiment error identification is a part of a primary resource scheduling process.

At 904, it is determined whether a conflict exists. If no conflict exists, a green light is displayed to the user at 908. If a conflict does exist, it is determined at 909 whether the conflict is resource specific, for example, whether the conflict relates to a particular employee. If the conflict is resource specific, it is determined at 910 whether the conflict is rule based or calendar based. If the conflict is rule base, a blue exclamation point is displayed at 911. If the conflict is calendar base, a red exclamation point is displayed at 913.

Referring again to block 909, if it is determined the conflict is not resource specific, it is then determined at 914 whether the conflict has been previously suppressed. If the conflict is suppressed, it is then determined at 915 whether there are other conflicts that are unsuppressed. If there are other conflicts that are unsuppressed, both a red and a yellow light are displayed at 918. If there are no unsuppressed conflicts, a green light is displayed at 920.

Referring again to 914, if the conflict is not suppressed, a red light is displayed at 916.

When a red, green or yellow light is displayed, the user can then click on the light for further information. If the user clicks on the red light, a new window containing an HTML page is generated describing the conflict and suggesting potential solutions. To do this, one embodiment utilizes a warning template file and the Windows registry.

If the user clicks on the yellow indication light, an HTML page is generated describing the suppressed conflict or conflicts and potential solutions. Contained within the new HTML page is a hyperlink that connects the user to the specific window within the primary scheduling resource software where the conflicted resource can be altered. If the user clicks on the hyperlink, one embodiment communicates with the primary resource scheduling software, causing the display of the specific window necessary to modify the data causing the error. Additionally, the data that should be altered is highlighted to facilitate resolving the conflict. If the user elects to suppress the conflict, the conflict is marked appropriately and stored in the Windows registry. If the user elects to un-suppress the conflict, an HTML page is generated based on the new unsuppressed conflict.